A complete transformation rule set and a minimal equation set for CNOT-based 3-qubit quantum circuits (Draft)

Issei Sakashita

May 8, 2013

Abstract. We introduce a complete transformation rule set and a minimal equation set for controlled-NOT (CNOT)-based quantum circuits. Using these rules, quantum circuits that compute the same Boolean function are reduced to a same normal form. We can thus easily check the equivalence of circuits by comparing their normal forms. By applying the Knuth-Bendix completion algorithm to a set of modified 18 equations introduced by Iwama et al. 2002 [IKY02], we obtain a complete transformation rule set (i.e., a set of transformation rules with the properties of 'termination' and 'confluence'). Our transformation rule set consists of 114 rules. Moreover, we found a minimal subset of equations for the initial equation set.

Keywords. Quantum circuit, String rewriting system

1. Introduction

Quantum computers were proposed in the early 1980s [Ben80, Ben82]. Significant contributions to quantum algorithms include the Shor factorization algorithm [Sho94, Sho97] and the Grover search algorithm [Gro96]. The quantum-circuit model of computation is due to Deutsch [Deu89], and it was further developed by Yao [Yao93].

After the works of Deutsch and Yao the concept of a universal set of quantum gates became central in the theory of quantum computation. A set $G = \{G_{1,n_1}, \cdots, G_{r,n_r}\}$ of r quantum gates G_{j,n_j} acting on n_j qubits ($j = 1, \cdots, r$), is called universal if any unitary action U_n on n input quantum states can be decomposed into a product of successive actions of G_{j,n_j} on different subsets of the input qubits [GMD02]. A first example of 3 qubit universal gate set consists of Deutsch's gates $\mathbf{Q}$ [Deu89]. The gate $\mathbf{Q}$ is an extension of the Toffoli gate [Tof81]. DiVincenzo showed that a set of two-qubit gates is exactly universal for quantum computation [DiV95]. After the result of DiVincenzo, Barenco showed that a large subclass of two-qubit gates are universal, and moreover, that almost any two-qubit gate is universal [Bar95]. Barenco et al. showed that the set consisting of one-qubit gates and CNOT gates is universal [BBC$^+$95]. There have been a number of studies that investigate the number of gates for decomposing an any gate of n qubits in $U(2^n)$. For the universal set consisting of one-qubit gates and CNOT gates, the number of gates is $O(n^3 4^n)$ by Barenco et al. [BBC$^+$95]. Knill reduced this bound to $O(n4^n)$ [Kni95]. Most useful information about universal quantum gates can be obtained from a survey paper written by A. Galindo and M.A. Marin-Delgado [GMD02].

The design of a good quantum circuit plays a key role in the successful implementation of a quantum algorithm. For this reason, Iwama et al. presented transformation rules that transform any 'proper' quantum circuit into a 'canonical' form circuit [IKY02]. There is, however, no discussion about the minimal size of a quantum circuit. In this article, we formulate a quantum circuit as a string and then simplify the circuit by using string rewriting rules to investigate them formally. Since a string rewriting system can be analyzed by using a monoid, we require several properties about monoids and groups.

String rewriting systems simplify strings by using transformation rules, and they have played a major role in the development of theoretical computer science. Several studies of string rewriting systems have been investigated [BO93]. Let M be a monoid and T a submonoid of finite index in M. If T can be presented by a finite complete rewriting system, so M can [Wan98]. The problem of confluence is, in general, undecidable. Parkes et al. showed that the class of groups that have monoid presentations obtainable by finite special [λ]-confluent string rewriting systems strictly contains the class of plain groups [PS04]. The word problem for a finite string is, in general, undecidable. If R is a finite string rewriting system that are Noetherian and confluent, then the word problem is decidable [Boo82, OZ91]. Book considered the word problem for finite string rewriting systems in which the notion of 'reduction' is based on rewriting the string as a shorter string [Boo82]. He showed that for any confluent system of this type, there is a linear-time algorithm for solving the word problem. Using a technique developed in [Boo82], Book and Ó'Dúlaing [BO81] showed that there is a polynomial-time algorithm for testing if a finite string rewriting system

is confluent. Gilman [Gil79] considered a procedure that, beginning with a finite string rewriting system, attempts to construct an equivalent string rewriting system that is Noetherian and confluent, that is, a string rewriting system such that every congruence class has a unique 'irreducible' string. This procedure appears to be a modification of the completion procedure developed by Knuth and Bendix [KB70] in the setting of term-rewriting systems. Narendran and Otto [NO88] also contributed to this topic. Later, Kapur and Narendran [KN85] showed how the Knuth-Bendix completion procedure could be adapted to the setting of string rewriting systems.

We do not deal with the general theory whether string rewriting systems are decidable or undecidable. We introduce an idea to reduce the size of a quantum circuit by using a string rewriting system. Our string rewriting rules based on 18 equations introduced by Iwama et al. 2002 [IKY02]. The Iwama's equations can not be considered as a complete rewriting rules as it is. That is, it does not have properties of termination and confluence. We would like to obtain a complete transformation rule set (i.e., a set of transformation rules with the properties of 'termination' and 'confluence') for reducing a quantum circuits. Therefore, we apply the Knuth-Bendix completion algorithm to a modified 18 equations. We obtain our complete transformation rule set consisted of 114 rules. There are three major results that are obtained by our investigation, for 3 qubits, the length of normal form is at most 6, the number of normal form is 168. Furthermore, we found a minimal subset of equations. The number of the general quantum circuits to arbitrary n qubits is already known by research of Clifford groups. So CQC_3 is considerd as a subgroup of a Clifford group.

This article consists of as follows. In section 2, we describe formal definitions of a quantum circuit. We consider a circuit that consists of just CNOT gates on 3 qubits. In section 3, we prove several properties for string rewriting systems. In section 4, we define a quantum circuit rewriting system for 3 qubits and show several related properties about it. We show that the number of normal forms is 168 on 3 qubits. In section 5, we found a minimal subset of equations.

2. Definitions of Quantum Circuits

In this section, we introduce several definitions related to quantum circuits. First, we define quantum bits (qubits), quantum gates, and quantum circuits.

Definition 1 (Quantum bits, gates, and circuits)**.** Let $\alpha, \beta \in \mathbb{C}, |0\rangle = (1,0), |1\rangle = (0,1)$ and $m \in \mathbb{N}$.

- A single qubit is denoted by a vector $|x\rangle = \alpha|0\rangle + \beta|1\rangle$.
- A n-qubit is denoted by $|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \in \mathbb{C}^{2^n}$.
- A n-qubit quantum gate is an unitary operator $G : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$.
- A quantum circuit Cir of size m is denoted by $Cir = (G_1, G_2, \cdots, G_m)$ where G_i $(i = 1, 2, \cdots, m)$ are n-qubit quantum gates.
- An empty circuit is denoted by λ.
- The output of circuit $Cir = (G_1, G_2, \cdots, G_m)$ for an input $|x\rangle$ is $(G_m \circ \cdots \circ G_2 \circ G_1)|x\rangle$.

Definition 2. Let $m, l \in N$, $Cir_1 = (G_1, G_2, \cdots, G_m)$ and $Cir_2 = (G'_1, G'_2, \cdots, G'_l)$ be n-qubit quantum circuits. We define an equivalence relation $=_{cir}$ by

$$\begin{aligned} &Cir_1 =_{cir} Cir_2 \\ &\iff \forall |x\rangle \in C^2, (G_m \circ \cdots \circ G_1)|x\rangle = (G'_l \circ \cdots \circ G'_1)|x\rangle. \end{aligned}$$

Next, we introduce a quantum gate that plays an important role in proving the universality of a quantum circuit.

Definition 3. The n-qubit controlled-NOT (CNOT) gate is a unitary operator $[c,t]_n : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ $(c, t \in \{1, 2, \cdots, n\})$ defined by

$$\bigotimes_{i=1}^{n} |\delta_i\rangle \mapsto \bigotimes_{i=1}^{t-1} |\delta_i\rangle \otimes |\delta_t \oplus \delta_c\rangle \otimes \bigotimes_{i=t+1}^{n} |\delta_i\rangle.$$

We call c the *control bit* and t the *target bit*. We use a version of Feynmann's notation [Fey85] for diagrammatic representations of CNOT gates (cf. Figure 1). An example of 3 qubit quantum circuit is illustrated in Figure 2. Each gate is applied in turn from left to right to the n qubits.

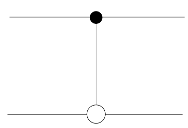

Figure 1: 2-qubit CNOT gate

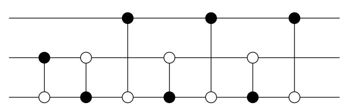

Figure 2: A quantum circuit

Next, we define an equivalence relation between two quantum circuits. This definition is important and allows us to discuss the equivalence of circuits. In this paper, we consider only quantum circuits that are constructed by 3-qubit CNOT gates. We denote as the set of circuits CQC_3 as

$$\begin{aligned} CQC_3 = \{&([c_1, t_1]_3, [c_2, t_2]_3, \cdots, [c_m, t_m]_3) | \\ &c_i, t_i \in \{1, 2, 3\}, c_i \neq t_i, m \in \mathbb{N}\}. \end{aligned}$$

We note that two different circuits Cir_1 and Cir_2 in CQC_3 , may be equivalent in the sense of $=_{cir}$, i.e. $Cir_1 =_{cir} Cir_2$.

Example 1. The following equation can be considered as illustrated in Figure.3:

$$([1,2]_3, [2,3]_3) =_{cir} ([2,3]_3, [1,2]_3, [1,3]_3).$$

For all input qubits $|\delta_1\rangle \otimes |\delta_2\rangle \otimes |\delta_3\rangle$, we need to prove the equivalence of the outputs. First, we compute $([1,3]_3 \circ [1,2]_3 \circ [2,3]_3)(|\delta_1\rangle \otimes |\delta_2\rangle \otimes |\delta_3\rangle)$,

$$\begin{aligned} & ([2,3]_3 \circ [1,2]_3)(|\delta_1\rangle \otimes |\delta_2\rangle \otimes |\delta_3\rangle) \\ =_{cir} \quad & [2,3]_3(|\delta_1\rangle \otimes |\delta_1 \oplus \delta_2\rangle \otimes |\delta_3\rangle) \\ =_{cir} \quad & |\delta_1\rangle \otimes |\delta_1 \oplus \delta_2\rangle \otimes |\delta_1 \oplus \delta_2 \oplus \delta_3\rangle. \end{aligned}$$

Next, we compute $([1,3]_3 \circ [1,2]_3 \circ [2,3]_3)(|\delta_1\rangle \otimes |\delta_2\rangle \otimes |\delta_3\rangle)$,

$$\begin{aligned} & ([1,3]_3 \circ [1,2]_3 \circ [2,3]_3)(|\delta_1\rangle \otimes |\delta_2\rangle \otimes |\delta_3\rangle) \\ =_{cir} \quad & ([1,3]_3 \circ [1,2]_3)(|\delta_1\rangle \otimes |\delta_2\rangle \otimes |\delta_2 \oplus \delta_3\rangle) \\ =_{cir} \quad & [1,3]_3(|\delta_1\rangle \otimes |\delta_1 \oplus \delta_2\rangle \otimes |\delta_2 \oplus \delta_3\rangle) \\ =_{cir} \quad & |\delta_1\rangle \otimes |\delta_1 \oplus \delta_2\rangle \otimes |\delta_1 \oplus \delta_2 \oplus \delta_3\rangle. \end{aligned}$$

Thus we have $([1,2]_3, [2,3]_3) =_{cir} ([2,3]_3, [1,2]_3, [1,3]_3)$.

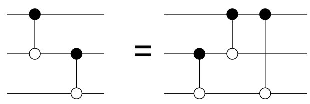

Figure 3: A circuit equation

We chose three types of simple equations to construct a string rewriting system.

Definition 4. Let G_1, G_2, and $G_3 \in CQC_3$ be CNOT gates.

- For any CNOT gate G, $(G, G) =_{cir} \lambda$ is an eliminated equation.
- $(G_1, G_2) =_{cir} (G_2, G_1)$ is a commutative equation.
- $(G_1, G_2) =_{cir} (G_2, G_1, G_3)$ is a anti-commutative equation .

In this article, we denote six CNOT gates for the 3 qubits $a = [1,2]_3$, $b = [1,3]_3$, $c = [2,1]_3$, $d = [2,3]_3$, $e = [3,1]_3$ and $f = [3,2]_3$.

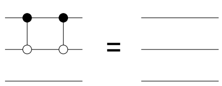

Figure 4: eliminated type: (a , a) = λ

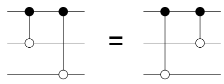

Figure 5: commutative type: (a , b) = (b , a)

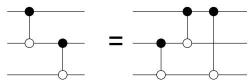

Figure 6: anti-commutative type: $(a, d) =_{cir} (d, a, b)$

3. String Rewriting System

In this section, we introduce the definition of a string rewriting system, in order to discuss quantum circuit transformation systems using it. Let Σ be a finite set of alphabets. We denote the set of all strings over Σ, including the empty string λ, as Σ^*. The length of a string $w \in \Sigma^*$ is denoted by $|w|$. A rewriting rule (u, v) is a pair of strings $u, v \in \Sigma^*$ where $u \neq \lambda$.

Definition 5 (string rewriting system)**.** A string rewriting system is a pair (Σ, R) of a finite set of alphabet Σ and a finite set of rewriting rules R .

Definition 6 (string rewriting)**.** Let (Σ, R) be a rewriting system and $s, t \in \Sigma^*$. We denote $s \to_R t$ if and only if there exist strings x, y, u and v in Σ^* such that $s = xuy, t = xvy$ and $(u, v) \in R$.
The reflexive transitive closure relation of $\to_R$ over Σ^* is denoted by $\to_R^*$. Further $\leftrightarrow_R^*$ is the symmetric closure relation of $\to_R^*$.

Definition 7 (irreducible, normal form)**.** Let (Σ, R) be a rewriting system and $w \in \Sigma^*$. For all substrings $c \subset w$, if there are no rules $(c, c') \in R$, then w is *irreducible*. For $s \in \Sigma^*$, if there exists s' such that $s \to_R^* s'$ and s' is irreducible, then s' is the $normal form$ of s. We denote the normal form s' of s as $NF(s)$.

The equivalence class of a string rewriting systems are considered using monoids, so we introduce several definitions and properties about monoids and their interprelations.

Definition 8 (monoid)**.** A monoid $M = (M, \cdot, \lambda)$ is a tuple of a set M, a binary operation $\cdot : M \times M \to M$, and a unit element $e \in M$ that satisfies the following two axioms.

- For any a, b, and c in $M, (a \cdot b) \cdot c = a \cdot (b \cdot c)$.
- For any a in M, $a \cdot \lambda = \lambda \cdot a = a$.

We note $(\Sigma^*, \cdot, \lambda)$ is a monoid where $\cdot$ is concatenation and λ is an empty string.

Definition 9 (homomorphism, isomorphic)**.** A homomorphism between two monoids $(M_1, \cdot_1, \lambda_1)$ and $(M_2, \cdot_2, \lambda_2)$ is

a function $f : M_1 \to M_2$ such that

- $f(x \cdot_1 y) = f(x) \cdot_2 f(y)$ for any $x, y \in M_1$, and
- $f(\lambda_1) = \lambda_2$.

If there exists a bijective homomorphism $f : M_1 \to M_2$, then M_1 and M_2 are isomorphic. We denote isomorphic as $M_1 \sim M_2$.

Proposition 1. *Let Σ be a finite set and $(M, \cdot, \lambda)$ a monoid. A function $f : \Sigma \to M$ is uniquely extended to the homomorphism $f^* : \Sigma^* \to M$ where $f^*(x_1 \cdot x_2 \cdots x_n) = f(x_1)f(x_2)\cdots f(x_n)$ and $f^*(\lambda) = \lambda$.*

□

Definition 10 (model, interpretation)**.** Let (Σ, R) be a rewriting system and $(M, \cdot, \lambda)$ a monoid. We say $(M, \cdot, \lambda)$ is a model of (Σ, R) if there exists a function $f : \Sigma \to M$ such that $f^*(u) = f^*(v)$ for any $(u, v) \in R$. We call the function f^* an interpretation of the string rewriting system (Σ, R) to a monoid M.

A string rewriting system can be investigated using a monoid and interpretation. We now add further definitions for discussing the equivalence of rewriting systems.

Definition 11 (factor monoid)**.** Let (Σ, R) be a rewriting system. A factor monoid $(\Sigma^*/R, \cdot, [\lambda])$ is defined by $\Sigma^*/R = \Sigma^*/\leftrightarrow_R^*$ and $[x] \cdot [y] = [xy]$ where $[x] = \{x' | x \leftrightarrow_R^* x'\}$

Proposition 2. *Let (Σ, R) be a rewriting system, $(M, \cdot, \lambda)$ a model of R and $f^* : \Sigma^* \to M$ an interpretation. The function $[f^*] : \Sigma^*/R \to M$ defined by $[f^*]([x]) = [f^*(x)](x \in \Sigma^*)$ is a homomorphism.* □

Definition 12 (rewriting system equivalence)**.** Let (Σ, R_1) and (Σ, R_2) be rewriting systems. R_1 and R_2 are equivalent if and only if Σ^*/R_1 and Σ^*/R_2 are isomorphic.

Finally, we introduce a lemma to compare two rewriting system that have the same alphabet Σ.

Lemma 1. *Let (Σ, R_1) and (Σ, R_2) be rewriting systems, and let $(M, \cdot, \lambda)$ be a monoid of (Σ, R_2). If there exists $(x_1, x_2) \in R_1$ and an interpretation $f : \Sigma^* \to M$ for Σ^*/R_2 such that $f^*(x_1) \neq f^*(x_2)$, then $\Sigma^*/R_1 \nsim \Sigma^*/R_2$.*

□

4. Quantum circuit rewriting system

We define a quantum circuit rewriting system for CQC_3.

Definition 13 (Quantum circuit rewriting system)**.** Let (Σ, R) be a string rewriting system and $i^* : \Sigma^* \to CQC_3/ =_{cir}$ a function where $\Sigma = \{a, b, c, d, e, f\}$, $i(a) = [1, 2]_3$, $i(b) = [1, 3]_3$, $i(c) = [2, 1]_3$, $i(d) = [2, 3]_3$, $i(e) = [3, 1]_3$ and $i(f) = [3, 2]_3$. (Σ, R) is a quantum circuit rewriting system, if i^* is an interpretation of (Σ, R). We identify a string $w = x_1x_2 \cdots x_n \in \Sigma^*$ as a circuit $(i(x_1), i(x_2), \cdots, i(x_n)) \in CQC_3$, and we also call w a *circuit*.

In general, string rewriting system do not have properties of 'termination' and 'confluence'. So we would like to construct a quantum circuit rewriting system that has both properties termination and confluence. To do so, we use the Knuth-Bendix algorithm [KB70, BO93, Met83].

Definition 14. Let E be an equation set. If the Knuth-Bendix algorithm succeeds for E, then we have a complete transformation rule set R (i.e., a set of transformation rules with the properties of termination and confluence). We denote $KBA(E)$ as the result of the Knuth-Bendix algorithm for E.

Example 2. Let A be an equation set s.t.

$$A = \begin{Bmatrix} aa = \lambda, & baba = abab, & dbd = bdb, \\ bb = \lambda, & dbabd = abab, & da = ad, \\ dd = \lambda & & \end{Bmatrix}.$$

We can compute $KBA(A)$:

$$KBA(A) = \begin{Bmatrix} aa \to \lambda, & ababd \to dbab, \\ abadb \to bdba, & abdba \to badb, \\ adbab \to babd, & baba \to abab, \\ babdb \to adba, & badba \to abdb, \\ bb \to \lambda, & bdbab \to abad, \\ da \to ad, & dbabd \to abab, \\ dbad \to bdba, & dbd \to bdb, \\ dd \to \lambda & \end{Bmatrix}.$$

Next, we apply the Knuth-Bedix completion algorithm to 18 equations

$$E_{all} = \begin{Bmatrix} aa = \lambda, & fbfb = a, & ab = ba \\ bb = \lambda, & adad = b, & bd = db \\ cc = \lambda, & dede = c, & cd = dc \\ dd = \lambda, & bcbc = d. & ce = ec \\ ee = \lambda, & fcfc = e, & af = fa \\ ff = \lambda, & eaea = f, & ef = fe \end{Bmatrix} \quad (1)$$

introduced by Iwama et. al. 2002 [IKY02]. We note that anti-commutative equations $xy = yxz$ (x, y and $z \in \Sigma$) equivalent to $xyxy = z$ ($xyxy = xyyxz = z$). We also call $xyxy = z$ (x, y and $z \in \Sigma$) anti-commutative equations. We used the *Mathmatica* software to compute the complete transformation rule set $KBA(E_{all})$, and we list it in the Appendix. The number of elements of $KBA(E_{all})$ is 114.

$$|KBA(E_{all})| = 114.$$

We note that we have applied an extended Knuth-Bendix algorithm which produce irreducible transformation rule set introduced in [Met83]. The transformation rule set $KBA(E_{all})$ is irreducible transformation rule set. The number of rules obtained by the original Knuth-Bendix algorithm is 244. A string rewriting system $(\Sigma, R_{E_{all}})$ is thus defined where $R_{E_{all}} = KBA(E_{all})$.

We would like to investigate commutativity of E_{all}.

Lemma 2. *We prove the following equations.*

1. $(acac, ca) \in R_{E_{all}}$,
2. $(bebe, eb) \in R_{E_{all}}$ *and*
3. $(dfdf, fd) \in R_{E_{all}}$.

Proof.

1. First, we show $fbca = caed$.

$$\begin{aligned} f(bc)a &= f(cbd)a \\ &= cfebda \\ &= cfebbad \\ &= cffaed \\ &= caed. \end{aligned}$$

Since $fbca = caed$ and $fbfb = a$,

$$\begin{aligned} acac &= (fbfb)cac \\ &= fb(caed)c \\ &= (caed)edc \\ &= ca(eded)c \\ &= cacc \\ &= ca. \end{aligned}$$

2. We can prove $bebe = eb$ by the same method to prove $acac = ca$. We rewrite $a \to b$, $b \to d$, $c \to e$, $d \to c$, $e \to f$ and $f \to a$ in the proof of $acac = ca$.

3. We can prove $dfdf = fd$ by the same method to prove $acac = ca$. We rewrite $a \to d$, $b \to c$, $c \to f$, $d \to e$, $e \to a$ and $f \to b$ in the proof of $acac = ca$.

□

Similarly, we obtain the following corollary.

Corollary 1.

1. $(caca, ac) \in R_{E_{all}}$,
2. $(ebeb, be) \in R_{E_{all}}$ *and*
3. $(fdfd, df) \in R_{E_{all}}$.

Proof.

1. Since $acac = ca$ and $cc = \lambda$, we have

$$\begin{aligned} caca &= caca(cc) \\ &= c(acac)c \\ &= c(ca)c \\ &= ac. \end{aligned}$$

2. 3. Similarly, we can prove.

□

By Lemma 2 and Corollary 1, we have the complete table of $xyxy$ for Σ^*/E_{all}.

Example 3. We show an equation $(ebe, beb) \in \Sigma/R_{E_{all}}$. Since $ebe = ebe(bb) = (ebeb)b = be$, we have $(ebe, beb) \in \Sigma/R_{E_{all}}$. We note that the rewriting rule $ebe \to beb$ appears on the last 6 line of Appendix.

Proposition 3. *Let $(\Sigma, R_{E_{all}})$ be a quantum circuit rewriting system where E_{all} a set of equations defined by (1). Then we have followings;*

1. $|NF(w)| \leq 6, (w \in \Sigma^7)$,
2. $|NF(w)| \leq 6, (w \in \Sigma^*)$, *and*
3. $|\Sigma^*/R_{E_{all}}| = 168$.

x \ y	a	b	c	d	e	f
a	λ	λ	ca	b	f	λ
b	λ	λ	d	λ	eb	a
c	ac	d	λ	λ	λ	e
d	b	λ	λ	λ	c	fd
e	f	be	λ	c	λ	λ
f	λ	a	e	df	λ	λ

Table 1: $xyxy$ for Σ^*/E_{all}

That is the length of $NF(w)$ is at most 6 for any string $w \in \Sigma^*$ and the number of normal forms is 168.

Proof.

1. We compute the $normalform$ for any string $w \in \Sigma^7$, then we have the length of $normalform$ is at most 6.

2. For any string $w \in \Sigma^*$ which length is $n \geq 7$, w contain a substring which length is 7. Thus w is rewritten to w' which length is at most $n - 1$. Inductively, for all string $w \in \Sigma^*$, the length of $NF(w)$ is at most 6.

3. We compute the $normalform$ for any string $w \in \Sigma^k$ $(1 \leq k \leq 6)$. So we have all members of $\Sigma^*/R_{E_{all}}$ and we have $|\Sigma^*/R_{E_{all}}| = 168$.

□

We list the all members of $\Sigma^*/R_{E_{all}}$ in Appendix. The question now arises: Is the set of equations wordy? Let E_6 be a set of equations such that

$$E_6 = E_{all} - \left\{ \begin{array}{c} ab = ba \\ bd = db \\ cd = dc \\ ce = ec \\ af = fa \\ ef = fe \end{array} \right\} = \left\{ \begin{array}{ll} aa = \lambda, & fbfb = a \\ bb = \lambda, & adad = b \\ cc = \lambda, & dede = c \\ dd = \lambda, & bcbc = d \\ ee = \lambda, & fcfc = e \\ ff = \lambda, & eaea = f \end{array} \right\}. \quad (2)$$

The size of this equation set is $|E_6| = 12$.

Lemma 3. *We prove the following equations.*

1. $(ba, ab) \in R_{E_6}$,
2. $(db, bd) \in R_{E_6}$,
3. $(dc, cd) \in R_{E_6}$,
4. $(ec, ce) \in R_{E_6}$,
5. $(fa, af) \in R_{E_6}$, *and*
6. $(fe, ef) \in R_{E_6}$.

Proof.

1. Since $bab = (adad)a(adad) = a$, we have $(ba, ab) \in R_{E_6}$.

2. 3. 4. 5. 6. We can prove similarly.

□

We compute a complete transformation rule set $KBA(E_6)$ by using the Knuth-Bendix algorithm, and we can have $KBA(E_6) = KBA(E_{all})$. The above results means that commutative type equations is not required for the initial equation set. We have next proposition.

Proposition 4. *Let (Σ, R) be a quantum circuit rewriting system, E_{all} and E_6 sets of equations defined by (1) and (2),*

$$\Sigma^*/R_{E_6} = \Sigma^*/R_{E_{all}}.$$

□

In the following section, we reduce the size of equation set and show the existence of the minimal set of equations E_{min} of E_6 that generates the isomorphic monoid $\Sigma^*/R_{E_{min}} = \Sigma^*/R_{E_6}$.

5. Minimal set of equations

Definition 15 (Minimal set of equations)**.** Let $E \subseteq \Sigma^* \times \Sigma^*$. A subset $E_{min} \subset E$ is a minimal equation set of E if and only if

- $\Sigma^*/R_{E_{min}} = \Sigma^*/R_E$, and
- If $\Sigma^*/R_{E'} = \Sigma^*/R_E$ then $|E_{min}| \leq |E'|$ for all $E' \subset E$.

In this section, we investigate a minimal set of equation of E_6 such that $\Sigma^*/R_{E_{min}} = \Sigma^*/R_{E_6}$. We delete some equations from E_6 and prove that the factor monoids of the equations are isomorphic. We follow the same line of thought as was used for the elementary Tietze transformation [BO93]. We first prove the following proposition.

Proposition 5. *Let (Σ, R) be a quantum circuit rewriting system, E_6 a set of equations defined by (2),*

$$E_5 = \left\{\begin{matrix} aa = \lambda, & fbfb = a \\ bb = \lambda, & adad = b \\ cc = \lambda, & dede = c \\ dd = \lambda, & bcbc = d \\ ee = \lambda, & fcfc = e \\ & eaea = f \end{matrix}\right\}, \textit{and}$$

$$E_2 = \left\{\begin{matrix} aa = \lambda, & fbfb = a \\ bb = \lambda, & adad = b \\ & dede = c \\ & bcbc = d \\ & fcfc = e \\ & eaea = f \end{matrix}\right\}. \tag{3}$$

Then we have followings:

1. $(efc, cf), ((fc)e, e(fc)) \in R_{E_5}$,
2. $(ff, \lambda) \in R_{E_5}$,
3. $\Sigma^*/R_{E_5} = \Sigma^*/R_{E_6}$, *and*
4. $\Sigma^*/R_{E_2} = \Sigma^*/R_{E_6}$.

Proof. We prove this proposition in following procedures.

1. Since

$$\begin{aligned} cf &= (ee)(aeaeeaea)cf(cc) \\ &= e(eaea)e(eaea)cfcc \\ &= efe(fcfc)c \\ &= efeec \\ &= efc, \end{aligned}$$

we have $(efc, cf) \in R_{E_5}$.
Since

$$\begin{aligned} fce &= fc(fcfc) \\ &= efc, \end{aligned}$$

we have $((fc)e, e(fc)) \in R_{E_5}$.

2. Since

$$\begin{aligned} ff &= f(cc)f \\ &= fc(efc) \\ &= (efc)fc \\ &= e(e) \\ &= \lambda, \end{aligned}$$

we have $(ff, \lambda) \in R_{E_5}$.

3. We show that $\Sigma^*/R_{E_5} = \Sigma^*/R_{E_6}$. Since $[ff]_{E_5} = [\lambda]_{E_5}$, we have $\Sigma^*/R_{E_5} = \Sigma^*/R_{E_6}$.

4. Let E_4, E_3 and E_2 be sets of equations where

$$\begin{aligned} E_4 &= E_6 - \{ee = \lambda, ff = \lambda\}, \\ E_3 &= E_6 - \{dd = \lambda, ee = \lambda, ff = \lambda\}, and \\ E_2 &= E_6 - \{cc = \lambda, dd = \lambda, ee = \lambda, ff = \lambda\}. \end{aligned}$$

Similarly, we have $(ee, \lambda) \in R_{E_4}$, $(dd, \lambda) \in R_{E_3}$ and $(cc, \lambda) \in R_{E_2}$. So we obtain

$$\begin{aligned} \Sigma^*/R_{E_6} &= \Sigma^*/R_{E_5} = \Sigma^*/R_{E_3} = \Sigma^*/R_{E_4} \\ &= \Sigma^*/R_{E_2}. \end{aligned}$$

□

Next, we prove that E_2 is a minimal set of equations.

Proposition 6.
Let $E' \subset E_6$. If $\Sigma^*/R_{E'} = \Sigma^*/R_{E_6}$, then $|E'| \geq 8$.

Proof. We will prove this in two stpdf.

step 1: First, we prove that we cannot remove a anti-commutative equation from E_6 .
We define a set of equations

$$E_{anti} = \left\{\begin{matrix} fbfb = a, adad = b, dede = c, \\ bcbc = d, fcfc = e, eaea = f \end{matrix}\right\}.$$

Let $u \in E_{anti}$ and consider u as $xyxy = z$ $(x, y, z \in \Sigma)$. We define E_u as $E_u = E_6 - \{u\} = E_6 - \{xyxy = z\}$.

Then we can show $\Sigma^*/R_{E_u} \neq \Sigma^*/R_{E_6}$ as follows. We consider a monoid $M = (\{0,1\}, \cdot, 0)$ where a binary operator $\cdot \subset \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined by Table 2, and a function $i : \Sigma^* \to M$ is defined as

$$i(\lambda) = i(k) = 0 (\forall k \neq z), \text{ and } i(z) = 1.$$

We consider a homomorphism i^* and show that i^* is an interpretation for E_u:

$$i^*(kk) = 0 \cdot 0 = 0 = i^*(\lambda),$$
$$i^*(zz) = 1 \cdot 1 = 0 = i^*(\lambda), \text{ and}$$

$$\begin{aligned} i^*(mnmn) &= i^*(mn) \cdot i^*(mn) \\ &= 0 \\ &= i^*(k), \forall m, n \in \Sigma, k \neq z. \end{aligned}$$

Since $x \neq z$ and $y \neq z$, then the value of $i^*(xyxy)$ is

$$i^*(xyxy) = i(0) \cdot i(0) \cdot i(0) \cdot i(0) = 0 \cdot 0 \cdot 0 \cdot 0 = 0.$$

Since $i^*(z)$ is

$$i^*(z) = i(z) = 1.$$

We have $i^*(xyxy) \neq i^*(z)$. By Lemma 1, $[xyxy]_{E_u} \neq [z]_{E_u}$. On the other hand, it is obvious that $[xyxy]_{E_6} = [z]_{E_6}$. Therefore,

$$\Sigma^*/R_{E_u} \neq \Sigma^*/R_{E_6}.$$

$\cdot$	0	1
0	0	1
1	1	0

Table 2: Definition of the binary operator $\cdot$

step 2: Let $x \in \Sigma$ and E_x a set of equations defined by

$$E_x = \left\{ \begin{array}{ll} xx = \lambda, & fbfb = a \\ & adad = b \\ & dede = c \\ & bcbc = d \\ & fcfc = e \\ & eaea = f \end{array} \right\}.$$

Then we can have $\Sigma^*/R_{E_x} \neq \Sigma^*/R_{E_6}$ as follows. For example, if we consider $x = a$, we can prove $\Sigma^*/R_{E_a} \neq \Sigma^*/R_{E_6}$. Let $N = (\{0,1,2\}, \cdot, 0)$ be a monoid where a binary operator $\cdot \subset \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined by Table 3, and let the function $j : \Sigma^* \to N$ be $j(\lambda) = j(a) = j(c) = 0, j(b) = j(e) = 1$ and $j(d) = j(f) = 2$. We consider a homomorphism j^* and show that j^* is an interpretation for E_a:

$$j^*(aa) = j(a) \cdot j(a) = 0 = j^*(\lambda),$$

$$j^*(fbfb) = j^*(fb) \cdot j^*(fb) = 0 \cdot 0 = 0 = j^*(a),$$

$$j^*(adad) = j^*(ad) \cdot j^*(ad) = 2 \cdot 2 = 1 = j^*(b),$$
$$j^*(dede) = j^*(de) \cdot j^*(de) = 0 \cdot 0 = 0 = j^*(c),$$
$$j^*(bcbc) = j^*(bc) \cdot j^*(bc) = 1 \cdot 1 = 2 = j^*(d),$$
$$j^*(fcfc) = j^*(fc) \cdot j^*(fc) = 2 \cdot 2 = 1 = j^*(e) \text{and}$$
$$j^*(eaea) = j^*(ea) \cdot j^*(ea) = 1 \cdot 1 = 2 = j^*(d).$$

The value of $j^*(bb)$ is

$$j^*(bb) = j(b) \cdot j(b) = 1 \cdot 1 = 2.$$

The value of $j^*(\lambda)$ is

$$j^*(\lambda) = j(\lambda) = 0.$$

Since $j^*(bb) \neq j^*(\lambda)$, we have $[bb]_{E_a} \neq [\lambda]_{E_a}$ by Lemma 1 . On the other hand, it is obvious that $[bb]_{E_6} = [\lambda]_{E_6}$. Therefore,

$$\Sigma^*/R_{E_a} \neq \Sigma^*/R_{E_6}.$$

$\cdot$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Table 3: Definition of the binary operator $\cdot$

Let $E' \subset E_6$ be a set of equations. If $\Sigma^*/R_{E'} = \Sigma^*/R_{E_6}$, it contained at least six anti-commutative equations by step1 and at least two eliminated equations by step2. Therefore, if $\forall E' \subset E_6$ it holds that $\Sigma^*/R_{E'} = \Sigma^*/R_{E_6}$, then

$$|E'| \geq 8.$$

□

Lemma 4. *Let E_{ac} be a set of equations where*

$$E_{ac} = \left\{ \begin{array}{ll} aa = \lambda, & fbfb = a \\ & adad = b \\ cc = \lambda, & dede = c \\ & bcbc = d \\ & fcfc = e \\ & eaea = f \end{array} \right\}.$$

Then

$$\Sigma^*/R_{E_{ac}} \neq \Sigma^*/R_{E_6}. \tag{4}$$

Proof. We show $[bb]_{E_{ac}} \neq [\lambda]_{E_{ac}}$. Let $N = (\{0,1,2\}, \cdot, 0)$ be a monoid where a binary operator $\cdot \subset \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined by Table 3, and let the function $j : \Sigma^* \to N$ be $j(\lambda) = j(a) = j(c) = 0, j(b) = j(e) = 1$ and $j(d) = j(f) = 2$. The function j is the same function used in step 2 of Proposition 6. We consider a homomorphism j^* and j^* is an interpretation for E_a. We check that $j^*(cc) = j^*(\lambda)$.

$$j^*(cc) = j(c) \cdot j(c) = 0 = j^*(\lambda).$$

The value of $j^*(bb)$ is

$$j^*(bb) = j(b) \cdot j(b) = 1 \cdot 1 = 2.$$

The value of $j^*(\lambda)$ is

$$j^*(\lambda) = j(\lambda) = 0.$$

Since $j^*(bb) \neq j^*(\lambda)$, we have $[bb]_{E_{ac}} \neq [\lambda]_{E_{ac}}$ by Lemma 1. On the other hand, it is obvious that $[bb]_{E_6} = [\lambda]_{E_6}$. Therefore,

$$\Sigma^*/R_{E_{ac}} \neq \Sigma^*/R_{E_6}.$$

□

From the above discussion, we derive the next theorem.

Theorem 1. *There exists a minimal equation set E_{min} of E_6 such that $|E_{min}| = 8$.*

Proof. E_2 is a minimal equation of E_6 by Propositions 5 and 6. □

Next, we show that there is an 8 element set of equations $F_2 \nsubseteq E_6$ such that $\Sigma^*/R_{F_2} = \Sigma^*/R_{E_6}$.

Proposition 7. *Let $F_2 \nsubseteq E_6$ be a set of equations defined by*

$$F_2 = \left\{ \begin{matrix} aa = \lambda, & bfbf = a \\ bb = \lambda, & dada = b \\ & eded = c \\ & cbcb = d \\ & cfcf = e \\ & aeae = f \end{matrix} \right\}.$$

Then $\Sigma^*/R_{F_2} = \Sigma^*/R_{E_6}$.

Proof. Let F_{anti} and F_6 be sets of equations defined by

$$F_{anti} = \left\{ \begin{matrix} bfbf = a, & dada = b, & eded = c \\ cbcb = d, & cfcf = e, & aeae = f \end{matrix} \right\}, \text{ and}$$

$$F_6 = \left\{ \begin{matrix} aa = \lambda, & bfbf = a \\ bb = \lambda, & dada = b \\ cc = \lambda, & eded = c \\ dd = \lambda, & cbcb = d \\ ee = \lambda, & cfcf = e \\ ff = \lambda, & aeae = f \end{matrix} \right\}.$$

We can prove

$$\Sigma^*/R_{F_2} = \Sigma^*/R_{F_6} \tag{5}$$

by following the same method in Proposition 4. Since we can prove $bfbf = a, dada = b, eded = c, cbcb = d, cfcf = e$ and $aeae = f$ in F_6, we have

$$\Sigma^*/R_{F_6} = \Sigma^*/R_{E_6 \cup F_{anti}}. \tag{6}$$

Similarly, we can have

$$\Sigma^*/R_{E_6 \cup F_{anti}} = \Sigma^*/R_{E_6}. \tag{7}$$

By (5), (6), and (7), we have $\Sigma^*/R_{F_2} = \Sigma^*/R_{E_6}$. □

6. Conclusion & Future Work

We considered rewriting systems in order to reduce the size of quantum circuits. We compute a set of complete transformation rules using the Knuth-Bendix algorithm. We discovered that the length of the normal form of $w \in \Sigma$ is at most 6 and the number of $|\Sigma^*/R_{E_2}|$ is 168.

We found a minimal equation set E_2 of the set of equations E_6 such that $|E_2| = 8$. On the other hand, we could construct a set of 8 equations $F_2 \nsubseteq E_6$ such that $\Sigma^*/R_{E_6} = \Sigma^*/R_F$. At this time, we do not have any equation set E such that $\Sigma^*/R_{E_6} = \Sigma^*/R_E$ and $|E| < 8$. The calculation of Knuth-Bendix algorithm for a smaller equation set is not always faster. We may take more computation time for a smaller equation set. The computation time of our implementation of Knuth-Bendix algorithms take 150 seconds for E_{all}, 330 seconds for E_6 and 1200 seconds for E_5.

In this paper, we restricted the size of qubits, so as an area of further work, we need to investigate about 4 or more qubits quantum circuits. We tried to compute Knuth-Bendix algorithm for 4 qubit quantum circuit rewriting system, we could not obtain results of computations.. We assume the cause to be computer power or set of equations.

Acknowledgements

I would like to thank Professor Yoshihiro Mizoguchi for his valuable advice and encouragement during the course of this study. I am grateful to Professor Miguel A. Martin-Delgado for his helpful comments.

References

[Bar95] A. Barenco. A universal two-bit gate for quantum computation. *arXiv:quant-ph/9505016*, May 1995.

[BBC+95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, November 1995.

[Ben80] P. Benioff. The comuter as a physical system: A microscopic quantum mechanical hamiltonian model of comuputers as represented by turing machines. *Journal of statistical physics*, 22(5):563–591, 1980.

[Ben82] P. Benioff. Quantum mechanical models of turing machines that dissipate no energy. *Physical Review Letters*, 48(23):1581–1585, March 1982.

[BO81] R. V. Book and C. P. O'Dunlaing. Testing for the church-rosser property. *Theoritical Computer Science*, 16:223–229, 1981.

[BO93] R. V. Book and F. Otto. *String rewriting system.* Springer-Verlag, 1993.

[Boo82] R. V. Book. Confluent and other types of thue systems. *Jornal of the ACM*, 29:171–182, January 1982.

[Deu89] D. Deutsch. Quantum computational networks. *Series A-Mathematical and Physical Sciences.*, 425(1868):73–90, Sept 1989.

[DiV95] D. P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51(2):1015–1022, 1995.

[Fey85] R. P. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985.

[Gil79] R. H. Gilman. Presentations of group and monoids. *Journal of Algebra*, 57:544–554, 1979.

[GMD02] A. Galindo and M. A. Martin-Delgado. Information and computation: Classical and quantum aspects. *Reviews of Modern Physics*, 74:347–423, Applil 2002.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. *symposium on Theory of computing*, 28:212–219, 1996.

[IKY02] K. Iwama, Y. Kambayashi, and S. Yamashita. Transformation rules for designing cnotbased quantum circuits. *Design Automation Conference*, 39th:419–424, 2002.

[KB70] D. Knuth and P. Bendix. Simple word problems in universal algebras. *Computational Problems in Abstract Algebra*, pages 263–297, 1970.

[KN85] D. Kapur and P. Narendran. A finite thue system with decidable word problem and without equivalent finite cononical system. *Theoritical Computer Science*, pages 337–344, 1985.

[Kni95] E. Knill. Approximation by quantum circuits. *arXiv:quant-ph/9508006*, 1995.

[Met83] Y. Metivier. About the rewriting system produced by the knuth-bendix completion algorithm. *Information Processing Letters*, 16:31–34, 1983.

[NO88] P. Narendran and F. Otto. Elements of finite order for finite weight-reducing and confluent thue systems. *Acta Informatica*, 25(5):573–591, June 1988.

[OZ91] F. Otto and L. Zhang. Decision problems for finite special string-rewriting systems that are confluent on some congruence class. *Acta Informatica*, 28:477–50, 1991.

[PS04] D. W. Parkes and V. Yu. Shavrukov. Monoid presentations of groups by finite special string-rewritings systems. *RAIRO-Inf. Theor. Appl.*, 38:245–256, 2004.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Foundations of Computer Science*, 35th:124–134, Nov 1994.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Tof81] Tommaso Toffoli. Bicontinuous extensions of inverrible combinatorial functions. *Mathematical Systems Theory*, 14:13–23, 1981.

[Wan98] J. Wang. Finite complete rewriting systems and finite derivtion type for small extensions of monoids. *Journal of Algebra*, 204:493–503, 1998.

[Yao93] A Chi-Chih Yao. Quantum circuit complexity. *Foundations of Computer Science*, 34th:352–361, 1993.

Appendix

We show the ruslut KBA(E_{all}) of Kunth-Bendix algorithm for E_{all} and the monoid $\Sigma^*/R_{E_{all}}$.

KBA$(E_{all}) = \{$ $aa \to \lambda$, $abcae \to caed$, $abcfbd \to beabc$, $abcfd \to fbde$, $abd \to da$, $abea \to bef$, $abef \to bea$, $abf \to fb$, $acaed \to bcae$, $acfbd \to eabc$, $ada \to bd$, $adfcb \to dfcd$, $adfcd \to dfcb$, $aea \to ef$, $aef \to ea$, $afb \to bf$, $ba \to ab$, $bb \to \lambda$, $bcab \to cda$, $bcaeb \to afcda$, $bcaed \to acae$, $bcb \to cd$, $bcd \to cb$, $bceab \to adefd$, $bcead \to adefb$, $bda \to ad$, $bdea \to adef$, $bdef \to adea$, $bfb \to af$, $bfcb \to afcd$, $bfcd \to afcb$, $cabc \to acda$, $cabeb \to bebdf$, $cabed \to acabe$, $cac \to aca$, $cad \to bca$, $caebd \to fcda$, $caeda \to fbc$, $cafc \to acea$, $cbc \to bd$, $cbd \to bc$, $cbe \to bed$, $cbfc \to adea$, $cc \to \lambda$, $cdaeb \to ebdf$, $cdaf \to bcfb$, $cde \to ed$, $cdfc \to def$, $ceabc \to acbfd$, $cebd \to deb$, $ced \to de$, $cef \to fc$, $cfbc \to aeda$, $cfbde \to beabc$, $cfc \to ef$, $dab \to ad$, $dac \to acb$, $dad \to ab$, $daebd \to abceb$, $daed \to abce$, $dafcb \to bdfcd$, $dafcd \to bdfcb$, $db \to bd$, $dc \to cd$, $dd \to \lambda$, $deab \to cead$, $dead \to ceab$, $debd \to ceb$, $ded \to ce$, $dfb \to adf$, $dfcda \to aebdf$, $eabca \to acafd$, $eabce \to bcfbd$, $eabcf \to beabc$, $eabe \to befb$, $eac \to acf$, $eade \to afcd$, $eadf \to dfcb$, $eae \to af$, $eaf \to ae$, $ebc \to deb$, $ebde \to bceb$, $ebdfc \to bebdf$, $ebdfd \to aebdf$, $ebe \to beb$, $ebf \to aeb$, $ec \to ce$, $edae \to bcfb$, $edaf \to cdae$, ede, cd, $edf \to dfc$, $ee \to \lambda$, $efbc \to cfbd$, $efbd \to aeda$, $efc \to cf$, $fa \to af$, $fbca \to caed$, $fbce \to abcf$, $fbcf \to abce$, $fbdeb \to beabc$, $fbdf \to dafd$, $fbe \to bea$, $fbf \to ab$, $fca \to cae$, $fcbf \to ceab$, $fcdae \to aebdf$, $fcdf \to defd$, $fce \to cf$, $fcf \to ce$, $fda \to bfd$, $fde \to cfd$, $fdf \to dfd$, $fe \to ef$, $ff \to \lambda$ $\}$

$\Sigma^*/R_{E_{all}} = \{\lambda$, a, ab, abc, $abca$, $abcaf$, $abcafd$, $abce$, $abcea$, $abceb$, $abcf$, $abcfb$, abe, $abeb$, $abebd$, $abebdf$, $abed$, $abeda$, ac, aca, $acab$, $acabe$, $acae$, $acaeb$, $acaf$, $acafd$, acb, $acbf$, $acbfd$, acd, $acda$, $acdae$, $acdf$, $acdfd$, ace, $acea$, $aceab$, $acead$, $aceb$, acf, $acfb$, $acfd$, ad, ade, $adea$, $adeb$, $adef$, $adefb$, $adefd$, adf, $adfc$, $adfd$, ae, aeb, $aebd$, $aebdf$, aed, $aeda$, af, afc, $afcb$, $afcd$, $afcda$, afd, b, bc, bca, $bcae$, $bcaf$, $bcafd$, bce, $bcea$, $bceb$, bcf, $bcfb$, $bcfbd$, $bcfd$, bd, bde, $bdeb$, bdf, $bdfc$, $bdfcb$, $bdfcd$, $bdfd$, be, bea, $beab$, $beabc$, $bead$, beb, $bebd$, $bebdf$, bed, $beda$, bef, $befb$, $befd$, bf, bfc, bfd, c, ca, cab, $cabe$, cae, $caeb$, $caed$, caf, $cafd$, cb, cbf, $cbfd$, cd, cda, $cdae$, cdf, $cdfd$, ce, cea, $ceab$, $cead$,

ceb, cf, cfb, $cfbd$, cfd, d, da, dae, $daeb$, daf, $dafc$, $dafd$, de, dea, deb, def, $defb$, $defd$, df, dfc, $dfcb$, $dfcd$, dfd, e, ea, eab, $eabc$, ead, eb, ebd, $ebdf$, ed, eda, ef, efb, efd, f, fb, fbc, fbd, $fbde$, fc, fcb, fcd, $fcda$, $fd\}$.

Issei Sakashita
Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
E-mail: i-sakashita@math.kyushu-u.ac.jp